REMARKS

Applicant is in receipt of the Office Action mailed March 3, 2009. Claims 82-102 are pending in the case. Reconsideration of the present case is earnestly requested in light of the following remarks.

Section 102 Rejection

Claims 82-102 were rejected under 35 U.S.C. 102(e) as being anticipated by Kudukoli, et al. (US Pat. Pub. 2001/0024211 A1, "Kudukoli").

Claim 82 recites:

82. A computer readable memory medium comprising program instructions, wherein the program instructions are executable by a processor to:

display a function node in a graphical program on a display, wherein the graphical program comprises a plurality of nodes and connections between the plurality of nodes, wherein the plurality of connected nodes visually indicate functionality of the graphical program, and wherein the function node is executable in the graphical program to perform a first function;

display a function specific property node in the graphical program on the display, wherein the function specific property node is specific to the first function, wherein the function specific property node comprises a plurality of properties of the first function;

associate the function specific property node with the function node;

display the plurality of properties on the display; and

receive user input selecting one or more of the plurality of properties;

wherein the selected one or more properties are accessible during execution of the graphical program.

Nowhere does Kudukoli teach or suggest **display a function specific property node in the graphical program on the display, wherein the function specific property node is specific to the first function, wherein the function specific property node comprises a plurality of properties of the first function**, as recited in claim 82.

Cited paragraph [0027] discusses graphical program generating (GPG) programs for generating graphical programs, but makes no mention of a property node at all, nor, more specifically, a *function specific* property node. This text discloses user input to the property node (which is *not* function specific, but rather is typed to a broad class of functionality) specifying an element filter option, where the attributes displayed by the property node are filtered accordingly. Figure 18 discloses a generic property node for configuring properties of *any* function node. Note that the property node of Figure 18 and [0250] is referred to as "the Property Node", indicating that there is only one, which is, in fact, the case; i.e., this property node is generic to all function nodes, and thus, is not a "function specific property node", as claimed. This point is emphasized in [0250], which states: "The Property node sets (writes) or gets (reads) application, VI and VI object property information. Thus, the Property node may be used to create/modify a graphical program. For example, the property node may be used to change the color of a user interface object, etc.", and in [0252] which states: "The Property node may be used to access or affect a number of different properties, including VI (virtual instrument) and application properties."

Paragraph [0251] describes how a user may select a property from an invoked list of properties, which depends on the type of object wired to the property node input. Note, however, that this same property node is used for *any* object, and that only the properties displayed change based on the wired node. In other words, the property node itself is still generic to all function nodes. Paragraph [0277] again describes use of "the Property node" to set properties of the graphical program, and again, clearly indicates that there is but one Property node that applies to all objects, and thus is generic.

Cited Figure 24 illustrates a block diagram generated by the GPG program of Figure 25. 25A illustrates the GPG program that creates the graphical program of Figures 23 (front panel/GUI) and 24 (block diagram). As described, this GPG program includes the Property node disclosed by Kudukoli, which, as explained above, is generic to all function nodes, and is thus, not a function specific property node, as claimed.

Nowhere does Kudukoli describe the Property node as being *specific to a particular function*.

Thus, as explained previously, Kudukoli's Property node is *not* function specific, but rather is used for "broad classes of functionality", and so may have "hundreds or even thousands of possible attributes". Kudukoli and the present invention respectively present two different and distinct solutions to this problem–Kudukoli teaches using a broad function class property node, and filtering the properties displayed to the user based on input to the property node; whereas in claim 82, the property node *is itself function specific*, and more particularly, is *specific to the first function of the function node*, and so only has attributes *for that function*, which is not at all the same.

Applicant further notes that filtering the displayed attributes of Kudukoli's property node does not change the property node itself, but rather, just limits which attributes are displayed to the user for selection. For example, the user could subsequently invoke a different filter option to change the displayed list of attributes–this does not convert the property node to a function-specific property node.

Thus, Kudukoli fails to teach or suggest this feature of claim 82.

Thus, for at least the reasons provided above, Applicant submits that Kudukoli fails to teach or suggest all the features of claim 82, and so claim 82, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Independent claim 102 includes similar limitations as claim 82, and so the above arguments apply with equal force to this claim. Thus, claim 102, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. For example:

Regarding claim 83, Applicant submits that Kudukoli does not disclose **wherein the property node is statically typed to correspond to the function node**, as recited in this claim.

Cited paragraph [0253] mentions inputs and outputs of Kudukoli's generic Property node. Nowhere does Kuduloki indicate that the Property node is statically typed to correspond to a particular function node.

In direct contrast, the property node of claim 83 is statically typed *to correspond to the function node*, and so only includes attributes that are specific to the function of that function node. Note that since this node is *statically* typed to correspond specifically to the function node, the node cannot be subsequently configured to correspond to a different type of function node (and thus to have different properties or attributes). In other words, the type of the property node of claim 83 is permanently established prior to runtime to be specific to the first function, and cannot be dynamically typed.

Applicant respectfully notes that Kudukoli mentions VI references being statically bound to specific VIs, but makes no mention whatsoever of statically typing property nodes, nor more specifically, property nodes that are statically typed to correspond to a function node. As noted above, Kuduloki only discloses one Property node, which is generic to *all* function nodes.

Thus, Kudukoli fails to disclose this feature of claim 83.

As another example, Kudukoli also fails to teach **wherein the function specific property node visually indicates the association with the function node**, as recited in claim 84.

Cited paragraph [0250] discloses that the Property node of Kudukoli, which Applicant notes is generic to all function nodes, "may be used to create/modify a graphical program", e.g., "to change the color of a user interface object, etc." However, this text says nothing about the Property node visually indicating an association with a function node. Additionally, as explained above, Kudukoli's Property node is not function specific.

Thus, the cited art fails to teach or suggest all the features of claim 84.

Applicant also asserts that numerous other ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been

shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

Removal of the section 102 rejection of claims 82-102 is respectfully requested.

Section 103 Rejection

Claim 86 was rejected under 35 U.S.C. 103(a) as being unpatentable over Kudukoli in view of Washington et al. (US 2002/0196282 A1, "Washington"). Applicant respectfully traverses the rejection.

The Office Action admits that Kudukoli fails to teach **display one or more filtering options for available properties of the function node, wherein the available properties include the plurality of properties; and receive user input indicating a first filtering option of the one or more filtering options, wherein said displaying the plurality of properties is performed in accordance with the first filtering option**, as recited in claim 84, but asserts that Washington remedies this admitted deficiency of Kudukoli, citing paragraph [0169] and Figures 20 and 21.

Cited paragraph [0169] discloses filtering of data from a data acquisition (DAQ) operation, but makes no mention of *properties filtering*. In other words, the Washington citation is directed to data processing (filtering of acquired data from a measurement process), whereas claim 86 recites displaying filtering options for display of function node properties, and displaying the function node properties accordingly. Thus, this citation is not germane to claim 86. Similarly, cited Figure 20 illustrates user interface controls which may be placed on a front panel in order to interactively obtain a reference to a server program or a graphical program to edit, which is also no relevant to these claimed features. Finally, cited Figure 21 illustrates how a user may select graphical program objects to create or edit by choosing from hierarchical menus, but again, has nothing whatsoever to do with filtering function node properties for display to a user, and more specifically, via a function specific property node, which neither reference discloses, or even hints at.

Thus, the cited art fails to teach or suggest all the features of claim 86.

Removal of the section 103 rejection of claim 86 is earnestly requested.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-81000/JCH.

Also filed herewith are the following items:

☐ Request for Continued Examination

☐ Terminal Disclaimer

☐ Power of Attorney By Assignee and Revocation of Previous Powers

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 2009-04-23 JCH/MSW